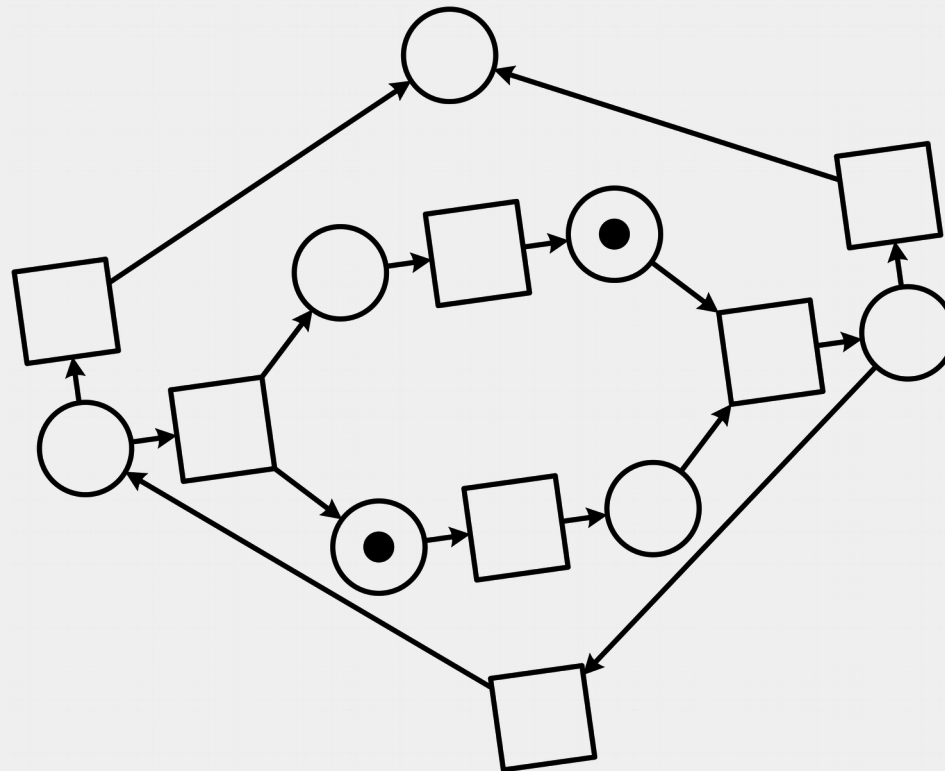


# ANALYSE VON PETRINETZMODELLEN



Matthias Weidlich



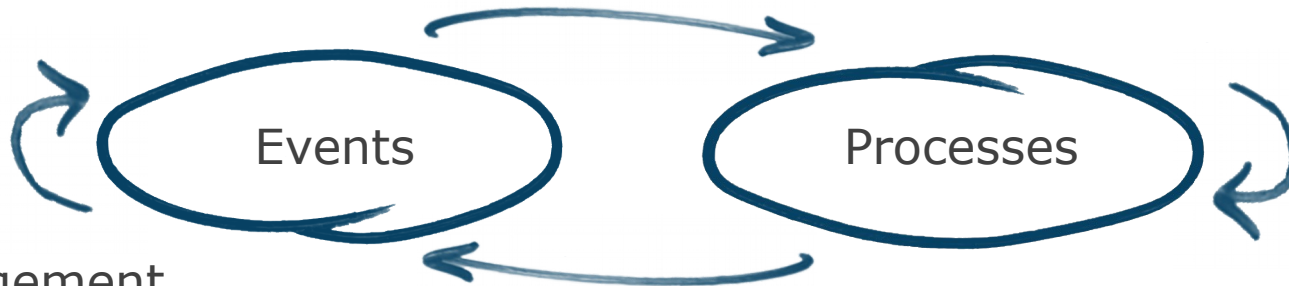
Process mining  
and prediction



Process equivalence  
and consistency



Uncertain  
data management



Event stream  
processing



Model matching and  
transformation

## Organisatorisches

- Planung/Anforderungen
- Spielregeln

## Einführung in das Seminarthema

- Modellierung
- Petrinetze

## Verteilung der Seminarthemen

# ORGANISATORISCHES

Mittwochs, 13 Uhr – c.t. oder s.t.?

Eine Seminarstunde: 90 Minuten

- Zweimal Vortrag (30 Min)
- Diskussion (15 Min)

16.04 Einführungveranstaltung

23.04 entfällt

30.04 – 12.02.14 Termine a 2 Vorträge

RUD 25, 4.112

Bachelor-Studierende Informatik

Vorkenntnisse

- Modellierung und Spezifikation?
- Software Engineering?
- Zeit und Petrinetze?

In AGNES eingeschrieben

## Vortrag

- 1x im Semester, 30 Minuten
- Selbständige Erarbeitung des Themas
- Einstiegsliteratur gegeben
- Mediennutzung: Beamer, Tafel

## Mitarbeit

- Beteiligung an inhaltlichen Diskussionen
- Beteiligung an der Vortragskritik

## Seminararbeit

- 10-15 Seiten
- Abgabezeitraum: Eigener Vortrag bis 31.03.2020
- Wissenschaftlicher Stil / „Probe-Artikel“

Vortrag vorher durchsprechen

Mit Medien vorher vertraut machen

Gut lesbare Schriftgröße

**Gut erkennbare Farben**

Bloß keine ganzen Sätze auf die Folien schreiben  
und die dann auch noch vorlesen!

Von übermäßigen Animationen absehen

Folien nummerieren

Gut strukturieren

Alle Begriffe einführen (Allg. bekannt vs. vermutlich  
bekannt vs. neu)

Viele Beispiele nutzen

Tipp: Erst Beispiel, dann Definition/Theorem



In jedem Vortrag: Positives und Negatives

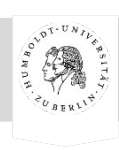
Immer erst positiv, dann negativ

Stets eigene Meinung von Fakten unterscheiden:

*Ich empfand die Struktur als sehr gelungen.  
Mir war zu viel Text auf den Folien.*

*Jeder neue Begriff wurde eingeführt.  
Die Folien waren nicht durchnummeriert.*

# **EINFÜHRUNG IN DAS SEMINARTHEMA**



# MODELLIERUNG

Modell: Abbildung eines realen Systems

Modellieren: Erzeugung eines Modells

Aspekte: **Verhalten** und Struktur

Analyse: Finden von Eigenschaften eines Modells

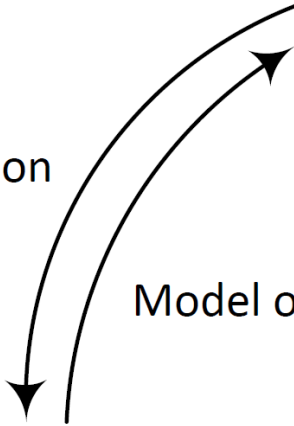
Verifikation: Nachweisen von Eigenschaften eines Modells

Original

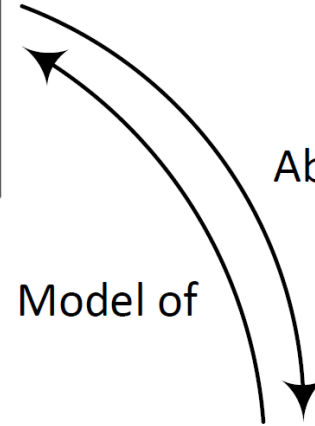


© National Aeronautics and Space Administration (NASA)

Abstraction



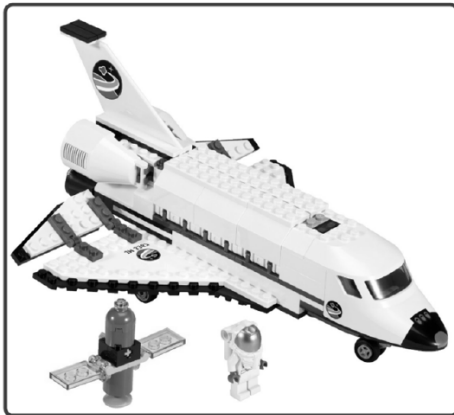
Model of



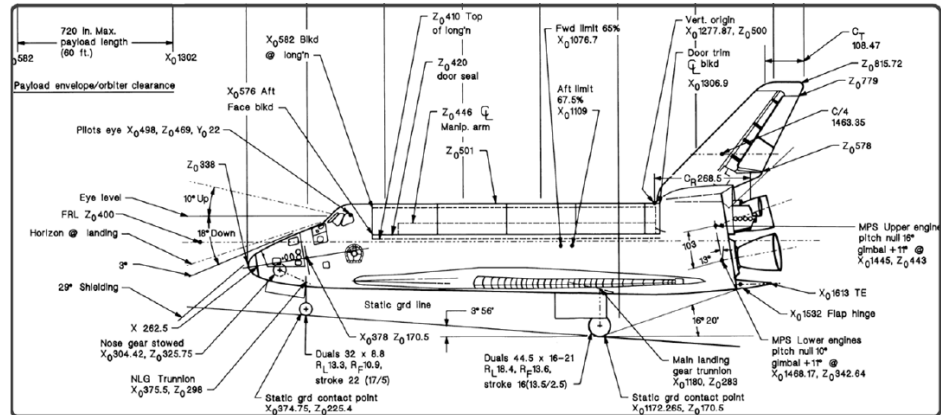
Abstraction

Model of

Model Level



© The LEGO Group



© Historic Space Systems at www.space1.com

Original



© Stiftung Deutsches Technikmuseum Berlin

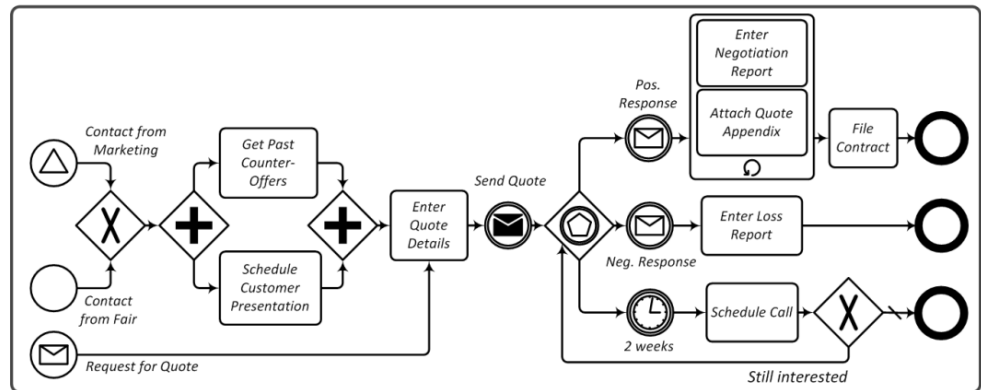
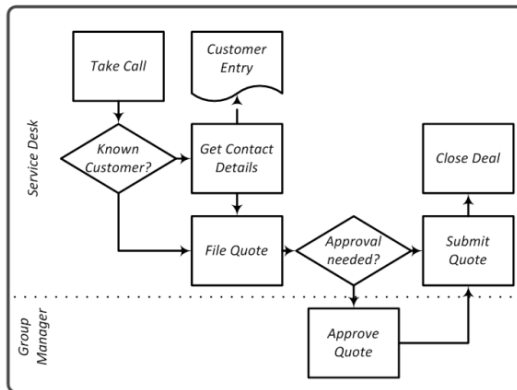
Abstraction

Abstraction

Model of

Model of

Model Level



Modellierung von...

- Zuständen (Variablen, Ressourcen, ...)
- Zustandsübergängen

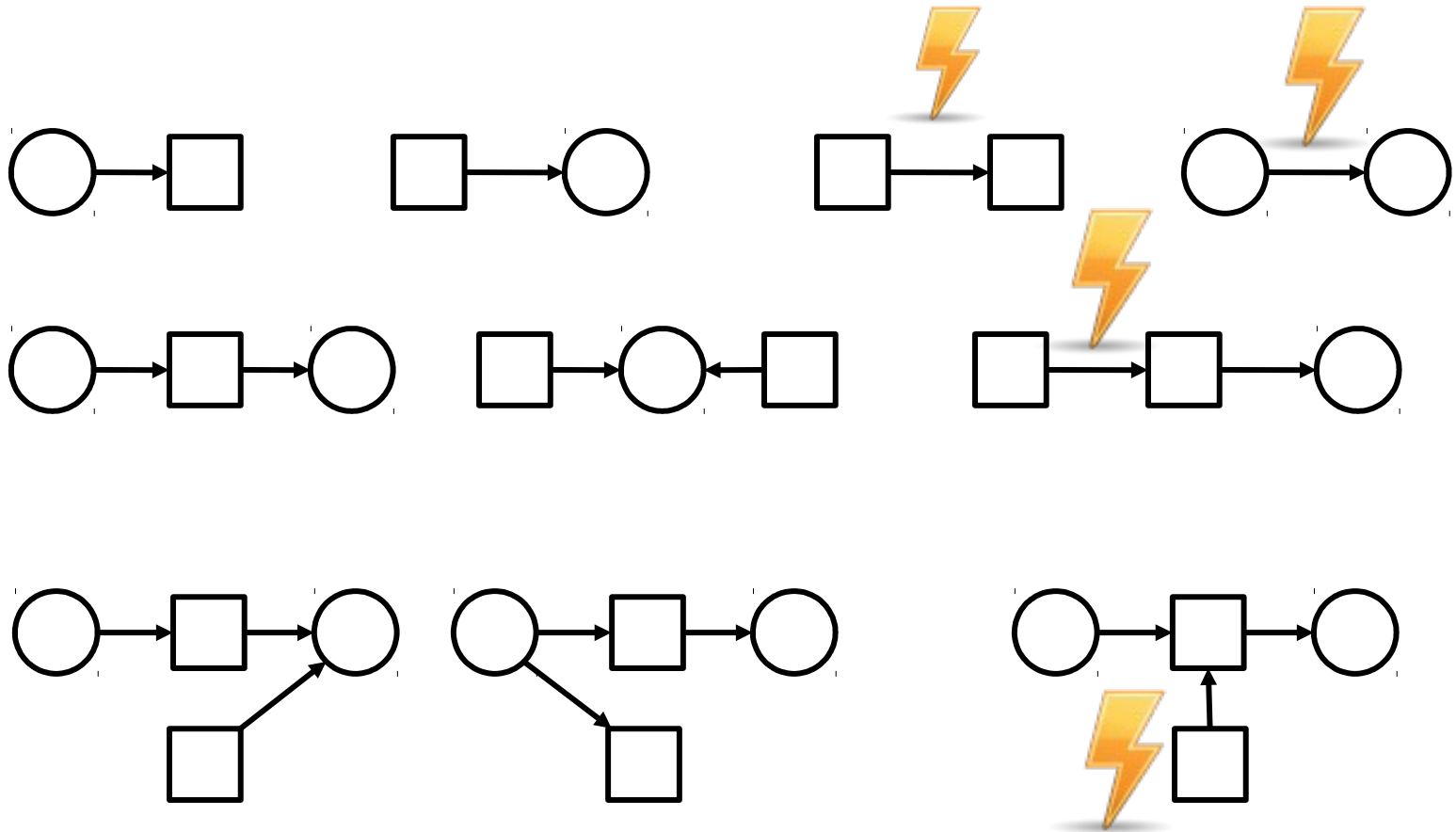
Zustandsautomaten, UML, Petrinetze, ...

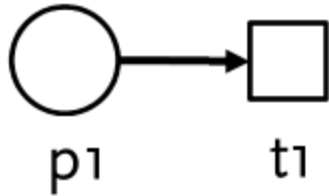
Petrinetzstrukturen

- Plätze/Stellen: Speicher/Bedingungen
- Transitionen: Aktionen
- Pfeile/Kanten



Pfeile verbinden Stellen mit Transitionen





$$\text{vor}(p_1) = \{ \}$$

$$\text{vor}(t_1) = \{p_1\}$$

$$\text{nach}(p_1) = \{t_1\}$$

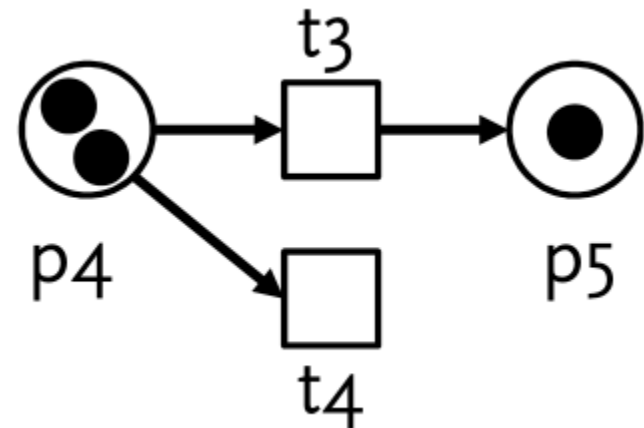
$$\text{nach}(t_1) = \{ \}$$



Marken (engl. Token) auf Stellen

Bedeutung abhängig von  
Bedeutung der Stelle

- Ressource
- Bedingung
- ...



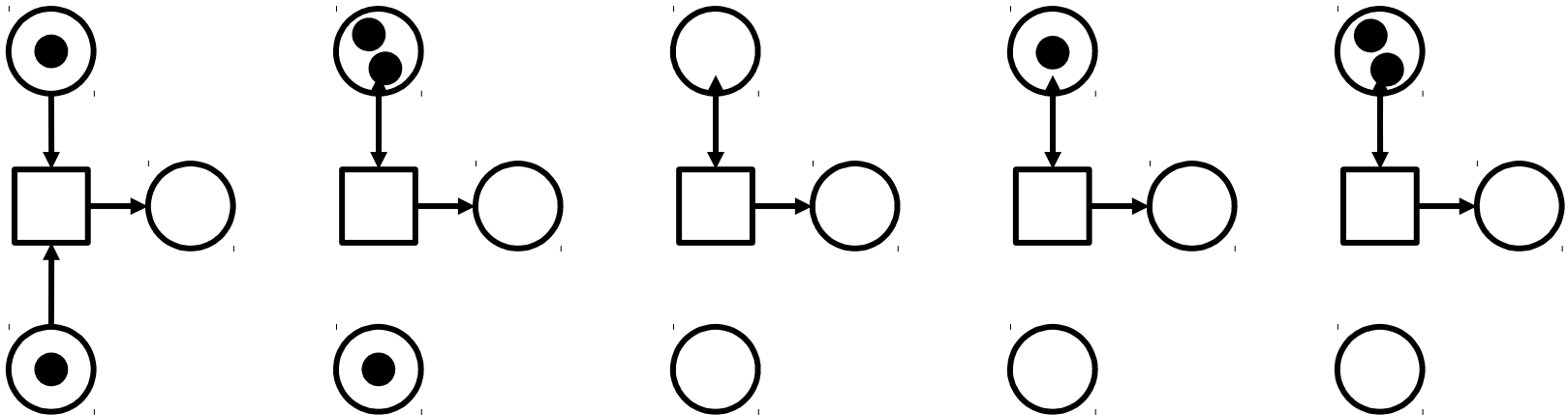
Aufzählung:  $[p4=2, p5=1]$

Vektor:  $(2,1)$

Multimenge:  $[p4, p4, p5]$

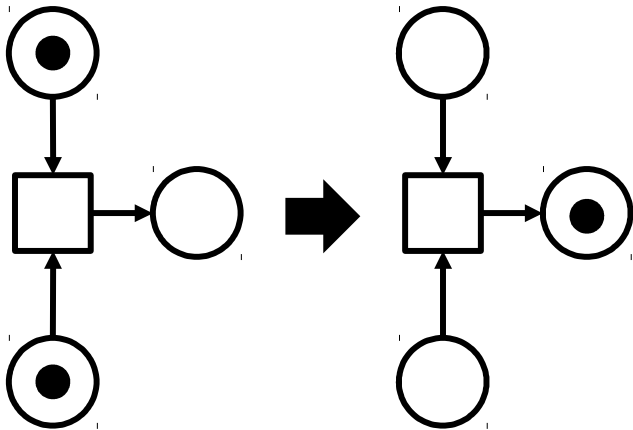
vor(t) markiert -> t ist **aktiviert**

- Alle Bedingungen erfüllt
- Alle Ressourcen vorhanden

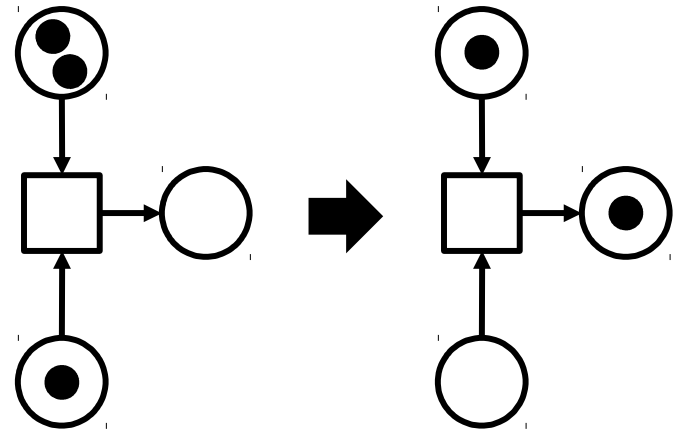


t ist aktiviert -> t **schaltet**

- Konsumieren von Token aus dem Vorbereich
- Produzieren von Token in den Nachbereich



$(1,1,0) \rightarrow (0,0,1)$



$(2,1,0) \rightarrow (1,0,1)$

Wenn  $t$  die Markierung  $M$  in Markierung  $M'$  überführt, heißt  $M \xrightarrow{t} M'$  **Schritt**

Folge von Schritten: sequenzieller Ablauf

Verhaltensmodellierung mit Petrinetzen

- Zustand: Markierung
- Zustandsübergang: Schritt

## Setting

- Vier Philosophen sitzen um einen Tisch
- Zwischen zwei Philosophen liegt jeweils eine Gabel

## Zustände/Regeln

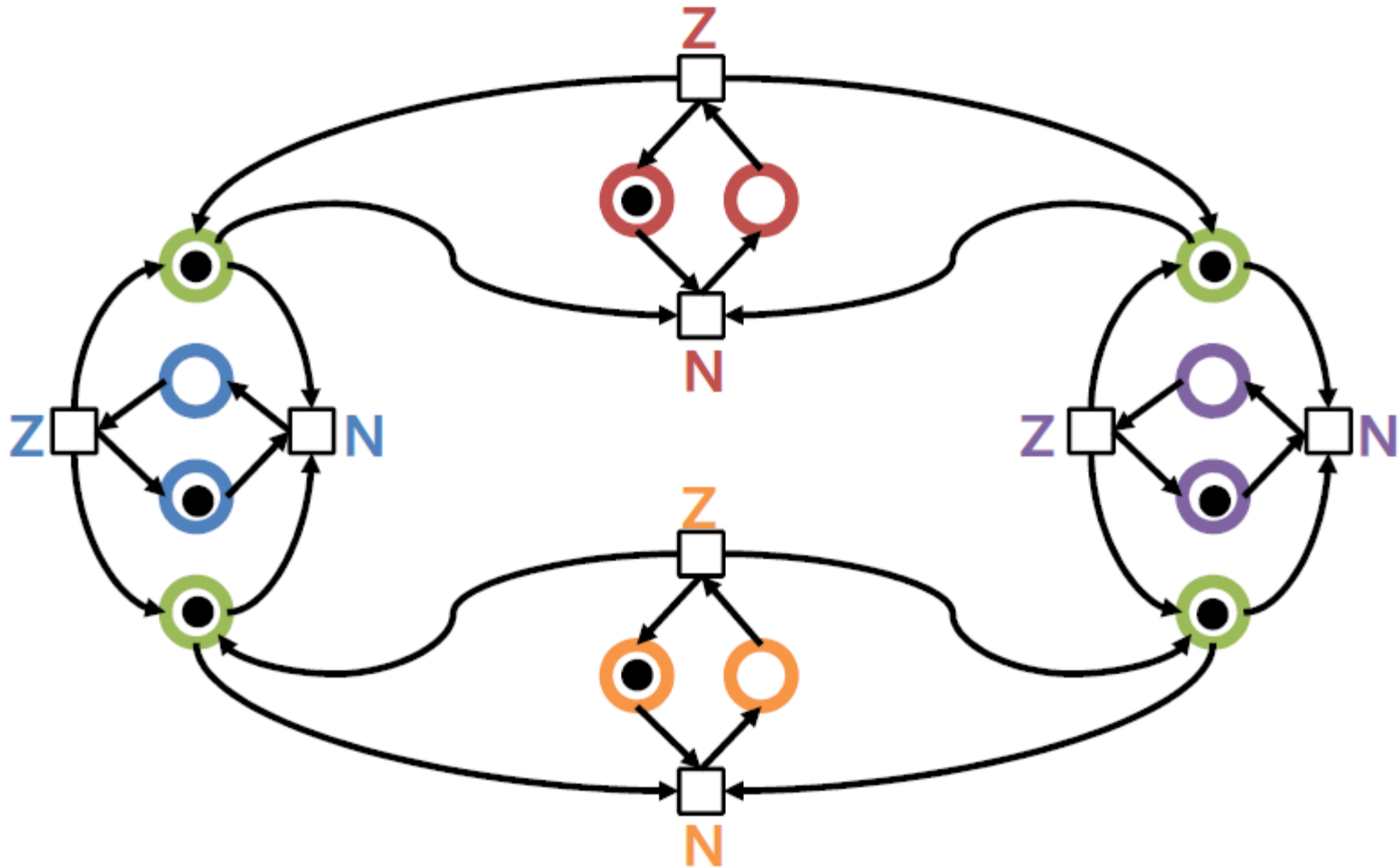
- Ein Philosoph kann immer nur essen oder denken – weder beides noch nichts davon.
- Ein Philosoph benötigt die rechte und die linke Gabel, um zu essen

## Ein Philosoph kann zwei Dinge tun:

- Beide Gabeln gleichzeitig nehmen
- Beide Gabeln gleichzeitig zurücklegen

## Initial:

- Alle Gabeln auf dem Tisch
- Alle Philosophen denken



Enthält alle von einem Anfangszustand erreichbaren Zustände

Notwendig: Anfangszustand -> **Anfangsmarkierung**

## **Markierungsgraph:**

- Gerichteter Graph
- Ein Knoten = eine Markierung
- Eine Kante = eine Transition
- Flussrelation = Schritt

**Erreichbarkeit:**  $M'$  erreichbar aus  $M$ , wenn ein Pfad von  $M$  nach  $M'$  im Markierungsgraphen existiert

**b-Beschränkt:** Höchstens  $b$  Token auf Stelle

**Beschränkt:** Ex.  $b$ , so dass  $b$ -beschränkt  $\rightarrow$  Markierungsgraph endlich

**Lebendig:** F.a.  $M$  und  $t$ : Ex.  $M'$ , so dass  $M'$  erreichbar aus  $M$  und  $t$  aktiviert in  $M'$

...



**Petrinetz:** 4-Tupel  $N = (P, T, F, M)$  mit

- Plätzen  $P$
- Transitionen  $T$
- $P, T$  disjunkt
- Flussrelation  $(P \times T) \cup (T \times P)$
- Anfangsmarkierung  $M: P \rightarrow \text{Nat}$

**Schritt:** Schalten einer Transition/Ein Übergang im Markierungsgraphen

- Sei  $t$  aktiviert in  $M$  und  $t$  überführe  $M$  in  $M'$
- Dann ist  $M \xrightarrow{t} M'$  ein Schritt

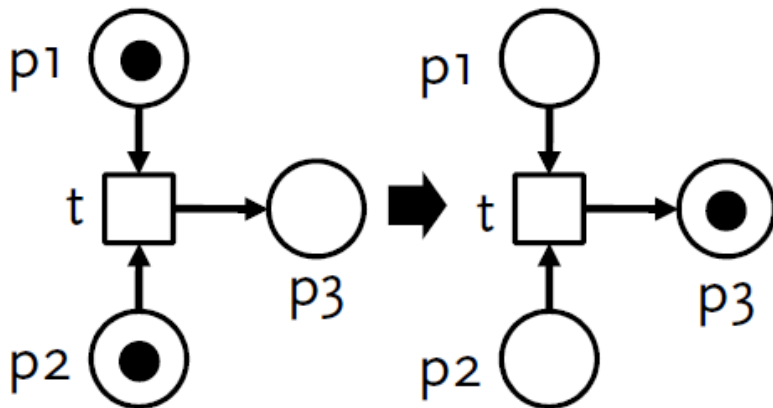
**Sequentieller Ablauf:** Sequenz von Schritten/Weg

im Markierungsgraphen

- Sei  $t_i$  aktiviert in  $M_i$  und  $t_i$  überführe  $M_i$  in  $M_{i+1}$
- Dann ist  $M_1 \xrightarrow{t_1} M_2 \xrightarrow{t_2} \dots$  ein sequenzieller Ablauf

## Rechnen mit Markierungen

- Markierung: Multimenge über P
- Sei t aktiviert in M, dann:  
 $M \xrightarrow{t} M - \text{vor}(t) + \text{nach}(t)$



$$\begin{aligned}
 [p1, p2] &\xrightarrow{t} \\
 [p1, p2] - \text{vor}(t) + \text{nach}(t) \\
 &= [p1, p2] - [p1, p2] + [p3] \\
 &= [p3]
 \end{aligned}$$

# THEMENVERGABE

1. Spezielle Netzklassen
2. Strukturelle Deadlocks und Fallen
3. Der Überdeckbarkeitsgraph
4. Platz- und Transitionsinvarianten
5. Verteilte Abläufe
6. Regionentheorie
7. Symmetrien
8. Partial Order Reduction
9. Unfoldings
10. Beweisgraphen
11. High-Level-Petrinetze
12. Zeit und Petrinetze
13. Stochastische Petrinetze
14. Workflow-Netze
15. High-level Invarianten
16. CPN-Tools
17. Vergleich von Werkzeugen für die Modellierung und Analyse
18. Die Sweep-Line-Methode für Petrinetze
19. Analyse von BPMN-Modellen
20. Komposition von Petrinetzen
21. Szenario-basierte Spezifikation mit Petrinetzen
22. Petrinetze und formale Sprachen
23. Stabile Ungleichungen in Petrinetzen
24. Verfeinerung in Petrinetzen